

CLAIMS:

1 1. A method for switching fonts without embedding font switches in the data
2 comprising the steps of:

3 receiving a character, wherein said character is a modified character in a first
4 base font resource or a character to be added to or deleted from said first base font
5 resource;

6 creating a font resource that comprises said character;

7 linking said created font resource to said first base font resource if said
8 character is a character to be added; and

9 linking said first base font resource to said created font resource if said
10 character is a character to be modified or deleted.

1 2. The method as recited in claim 1 further comprising the steps of:

2 creating an entry in a first table indicating said created font resource is a
3 second base font resource; and

4 creating a second and a third table associated with said created font resource,
5 wherein said second table maps code points to glyph indexes, wherein said third table
6 comprises glyphs

1 3. The method as recited in claim 2 further comprising the steps of:

2 creating a link list in an entry in said first table associated with said first base
3 font resource to link said created font resource to said first base font resource;

4 indicating in said entry in said first table associated with said first base font
5 resource to not reverse linking of said first base font resource to said created font
6 resource if said character is a character to be added; and

7 indicating in said entry in said first table associated with said first base font
8 resource to reverse linking of said first base font resource to said created font resource
9 if said character is a character to be modified or deleted.

1 4. The method as recited in claim 3 further comprising the steps of:
2 receiving an identification of a font resource and a code point; and
3 transmitting said code point to a rasterizer program associated with said
4 identified font resource.

1 5. The method as recited in claim 4, wherein said first base font resource is
2 associated with a fourth table and a fifth table, wherein said fourth table maps code
3 points to glyph indexes, wherein said fifth table comprises glyphs.

1 6. The method as recited in claim 5 further comprising the step of:
2 determining if said code point indexes in said fourth table;
1 wherein if said code point indexes in said fourth table, then the method further
2 comprises the steps of:
3 procuring a glyph from said fifth table using a glyph index obtained
4 from said fourth table;
5 converting said glyph to a bit map representation; and
6 transmitting said bit map representation to a printer; and
7 wherein if said code point does not index in said fourth table, then the method
8 further comprises the step of:
9 determining if said code point indexes in said second table.

1 7. The method as recited in claim 5 further comprising the step of:
2 determining if said code point indexes in said second table;
1 wherein if said code point indexes in said second table, then the method
2 further comprises the steps of:
3 procuring a glyph from said third table using a glyph index obtained
4 from said second table;
5 converting said glyph to a bit map representation; and
6 transmitting said bit map representation to a printer; and

- 7 wherein if said code point does not index in said second table, then the
8 method further comprises the step of:
9 determining if said code point indexes in said fourth table.

1 8. A computer program product embodied in a machine readable medium for
2 switching fonts without embedding font switches in the data comprising the
3 programming steps of:

4 receiving a character, wherein said character is a modified character in a first
5 base font resource or a character to be added to or deleted from said first base font
6 resource;

7 creating a font resource that comprises said character;

8 linking said created font resource to said first base font resource if said
9 character is a character to be added; and

10 linking said first base font resource to said created font resource if said
11 character is a character to be modified or deleted.

1 9. The computer program product as recited in claim 10 further comprising the
2 programming steps of:

3 creating an entry in a first table indicating said created font resource is a
4 second base font resource; and

5 creating a second and a third table associated with said created font resource,
6 wherein said second table maps code points to glyph indexes, wherein said third table
7 comprises glyphs

1 10. The computer program product as recited in claim 10 further comprising the
2 programming steps of:

3 creating a link list in an entry in said first table associated with said first base
4 font resource to link said created font resource to said first base font resource;

5 indicating in said entry in said first table associated with said first base font
6 resource to not reverse linking of said first base font resource to said created font
7 resource if said character is a character to be added; and

8 indicating in said entry in said first table associated with said first base font
9 resource to reverse linking of said first base font resource to said created font resource
10 if said character is a character to be modified or deleted.

1 11. The computer program product as recited in claim 10 further comprising the
2 programming steps of:

3 receiving an identification of a font resource and a code point; and
4 transmitting said code point to a rasterizer program associated with said
5 identified font resource.

1 12. The computer program product as recited in claim 11, wherein said first base
2 font resource is associated with a fourth table and a fifth table, wherein said fourth
3 table maps code points to glyph indexes, wherein said fifth table comprises glyphs.

1 13. The computer program product as recited in claim 12 further comprising the
2 programming step of:

3 determining if said code point indexes in said fourth table;
4 wherein if said code point indexes in said fourth table, then the computer
5 program product further comprises the programming steps of:

6 procuring a glyph from said fifth table using a glyph index obtained
7 from said fourth table;

8 converting said glyph to a bit map representation; and

9 transmitting said bit map representation to a printer; and

10 wherein if said code point does not index in said fourth table, then the
11 computer program product further comprises the programming step of:

12 determining if said code point indexes in said second table.

1 14. The computer program product as recited in claim 13 further comprising the
2 programming step of:

3 determining if said code point indexes in said second table;

4 wherein if said code point indexes in said second table, then the computer
5 program product further comprises the programming steps of:

6 procuring a glyph from said third table using a glyph index obtained
7 from said second table;
8 converting said glyph to a bit map representation; and
9 transmitting said bit map representation to a printer; and
10 wherein if said code point does not index in said second table, then the
11 computer program product further comprises the programming step of:
12 determining if said code point indexes in said fourth table.

1 15. A system, comprising:
2 a client configured to generate a first data stream comprising page description
3 information;
4 a spool coupled to said client, wherein said spool is configured to store said
5 first data stream;
6 a resource library configured to store a first base font resource;
7 a print server coupled to said spool and said resource library, wherein said
8 print server comprises:
9 a first memory unit operable for storing a printer driver configured to
10 generate a second data stream; and
11 a first processor coupled to said first memory unit; and
12 a printer coupled to said print server, wherein said printer is configured to
13 receive said second data stream generated from said print server, wherein said printer
14 comprises:
15 a second memory unit operable for storing a rasterizer program; and
16 a control unit coupled to said second memory unit;
17 wherein said client comprises:
18 a third memory unit operable for storing a computer program for
19 creating a linked resource;
20 a second processor coupled to said second memory unit, wherein said
21 second processor, responsive to said computer program, comprises:
22 circuitry operable for receiving a character, wherein said
23 character is a modified character in said first base font resource or a character to be
24 added to or deleted from said first base font resource;
25 circuitry operable for creating a font resource that comprises
26 said character;
27 circuitry operable for linking said created font resource to said
28 first base font resource if said character is a character to be added; and
29 circuitry operable for linking said first base font resource to
30 said created font resource if said character is a character to be modified or deleted.

1 16. The system recited in claim 15, wherein said resource library comprises a first
2 table, wherein said second processor further comprises:

3 circuitry operable for creating an entry in said first table indicating said
4 created font resource is a second base font resource; and

5 circuitry operable for creating a second and a third table associated with said
6 created font resource to be stored in said resource library, wherein said second table
7 maps code points to glyph indexes, wherein said third table comprises glyphs.

1 17. The system as recited in claim 16, wherein said resource library comprises a
2 first table, wherein said second processor further comprises:

3 circuitry operable for creating a link list in an entry in said first table
4 associated with said first base font resource to link said created font resource to said
5 first base font resource;

6 circuitry operable for indicating in said entry in said first table associated with
7 said first base font resource to not reverse linking of said first base font resource to
8 said created font resource if said character is a character to be added; and

9 circuitry operable for indicating in said entry in said first table associated with
10 said first base font resource to reverse linking of said first base font resource to said
11 created font resource if said character is a character to be modified or deleted.

1 18. The system as recited in claim 17, wherein said first processor, responsive to
2 said printer driver, comprises:

3 circuitry operable for receiving an identification of a font resource and a code
4 point; and

5 circuitry operable for transmitting said code point to said rasterizer program
6 associated with said identified font resource.

1 19. The system as recited in claim 18, wherein said resource library stores a fourth
2 table and a fifth table associated with said first base font resource, wherein said fourth
3 table maps code points to glyph indexes, wherein said fifth table comprises glyphs.

1 20. The system as recited in claim 19, wherein said control unit, responsive to
2 said rasterizer program, comprises:

3 circuitry operable for determining if said code point indexes in said second
4 table;

1 wherein if said code point indexes in said second table, then said control unit,
2 responsive to said rasterizer program, further comprises:

3 circuitry operable for procuring a glyph from said third table using a
4 glyph index obtained from said second table;

5 circuitry operable for converting said glyph to a bit map
6 representation; and

7 circuitry operable for transmitting said bit map representation to a
8 printer; and

9 wherein if said code point does not index in said second table, then said
10 control unit, responsive to said rasterizer program, further comprises:

11 circuitry operable for determining if said code point indexes in said
12 fourth table.

1 21. The system as recited in claim 19, wherein said control unit, responsive to
2 said rasterizer program, comprises:

3 circuitry operable for determining if said code point indexes in said fourth
4 table;

1 wherein if said code point indexes in said fourth table, then said control unit,
2 responsive to said rasterizer program, further comprises:

3 circuitry operable for procuring a glyph from said fifth table using a
4 glyph index obtained from said fourth table;

5 circuitry operable for converting said glyph to a bit map
6 representation; and

7 circuitry operable for transmitting said bit map representation to a
8 printer; and

9 wherein if said code point does not index in said fourth table, then said control
10 unit, responsive to said rasterizer program, further comprises:
11 circuitry operable for determining if said code point indexes in said
12 second table.

1 22. A method for switching fonts without embedding font switches in the data
2 comprising the steps of:

3 receiving an identification of a font resource;

4 searching in the inline data for a native name associated with said font
5 resource; and

6 searching a table for said native name associated with said font resource if
7 said native name is not located in said inline data.

1 23. The method as recited in claim 22 further comprising the steps of:

2 searching in said inline data for a link list associated with said font resource;
3 and

4 searching in said table for said link list associated with said font resource if
5 said link list is not located in said inline data;

6 searching for a native name associated with any linked font resources
7 identified in said link list in said inline data; and

8 searching for said native name associated with any linked font resources
9 identified in said link list in said table if said native name is not located in said inline
10 data.

1 24. A computer program product embodied in a machine readable medium for
2 switching fonts without embedding font switches in the data comprising the
3 programming steps of:

4 receiving an identification of a font resource;

5 searching in the inline data for a native name associated with said font
6 resource; and

7 searching a table for said native name associated with said font resource if
8 said native name is not located in said inline data.

1 25. The computer program product as recited in claim 24 further comprising the
2 programming steps of:

3 searching in said inline data for a link list associated with said font resource;
4 and

5 searching in said table for said link list associated with said font resource if
6 said link list is not located in said inline data;

7 searching for a native name associated with any linked font resources
8 identified in said link list in said inline data; and

9 searching for said native name associated with any linked font resources
10 identified in said link list in said table if said native name is not located in said inline
11 data.

1 26. A system, comprising:
2 a client configured to generate a first data stream comprising page description
3 information;
4 a spool coupled to said client, wherein said spool is configured to store said
5 first data stream;
6 a resource library configured to store font resources;
7 a print server coupled to said spool and said resource library, wherein said
8 print server comprises:
9 a memory unit operable for storing a printer driver configured to
10 generate a second data stream; and
11 a processor coupled to said first memory unit;
12 wherein said processor, responsive to said printer driver, comprises:
13 circuitry operable for receiving an identification of a font
14 resource;
15 circuitry operable for searching in said first data stream for a
16 native name associated with said font resource; and
17 circuitry operable for searching a table in said resource library
18 for said native name associated with said font resource if said native name is not
19 located in said first data stream.

1 27. The system as recited in claim 26, wherein said processor, responsive to said
2 printer driver, comprises:
3 circuitry operable for searching in said first data stream for a link list
4 associated with said font resource; and
5 circuitry operable for searching in said table in said resource library for said
6 link list associated with said font resource if said link list is not located in said first
7 data stream;
8 circuitry operable for searching for a native name associated with any linked
9 font resources identified in said link list in said first data stream; and

10 circuitry operable for searching for said native name associated with any
11 linked font resources identified in said link list in said table if said native name is not
12 located in said first data stream.